



Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Jia, Wenzheng, Mau, Baohua, [Ho, Tin Kin](#), Liu, Haidong, & Yang, Bo-Suk (2009) Bottlenecks detection of track allocation schemes at rail stations by Petri nets. *Journal of Transportation Systems Engineering and Information Technology*, 9(6), pp. 136-141.

This file was downloaded from: <http://eprints.qut.edu.au/38257/>

© Copyright 2009 China Association for Science and Technology

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

[http://dx.doi.org/10.1016/S1570-6672\(08\)60094-X](http://dx.doi.org/10.1016/S1570-6672(08)60094-X)

Bottlenecks Detection of Track Allocation Schemes at Rail Stations by Petri Nets

JIA Wen-zheng^{1,2}, MAO Bao-hua^{*1}, HO Tin-kin², LIU Hai-dong¹, YANG Bo³

(1 Ministry of Education Key Laboratory of Complex System Theory & Technology of Urban Traffic, Beijing Jiao tong University, Beijing 100044; 2 Hong Kong Polytechnic University, Department of Electrical Engineering, Hong Kong, China; 3 Nancang Station, Beijing Railway Bureau, Tianjin 300402, China)

Abstract: Robustness of the track allocation problem is rarely addressed in literatures and the obtained track allocation schemes (TAS) embody some bottlenecks. Therefore, an approach to detect bottlenecks is needed to support local optimization. First a TAS is transformed to an executable model by Petri nets. Then disturbances analysis is performed using the model and the indicators of the total trains' departure delays are collected to detect bottlenecks when each train suffers an disturbance. Finally, the results of the tests based on a rail station linking six lines and a TAS about 30 minutes show that the minimum buffer time is 21 seconds and there are two bottlenecks where the buffer times are 57 and 44 seconds respectively, and it indicates that the bottlenecks is not always located in the area where there is minimum buffer time. The proposed approach can further support TAS evaluation and robustness optimization.

key words: Rail Stations; Track Allocation Schemes; Bottlenecks Detection; Petri nets

CLC number: U292.18

Document code: A

1. Introduction

Routing trains at railway stations is a common problem in railway scheduling and operation [1-2]. The objective is to allocate conflict-free inbound & outbound routes and platforms to trains while ensuring the operations safety and achieving reasonable infrastructure utilization. Solutions to this track allocation problem (TAP) are referred to as track allocation schemes (TAS) in the paper. The objectives in the literatures are normally the maximum platforms preference, workload balance of the platform tracks and the minimum of shunting trains. Buffer times imply the delay-tolerance of a TAS; however they are rarely addressed in previous studies. Disturbances are inevitable in real-life operations and the TAS delay-tolerance is good if the sensitivity to disturbances is low, and the performance is referred to as robustness [3] which is an emerging issue in railway and air transport timetabling. The train scheduling in China mainland railways is reviewed in the literature [4] where an integrated approach is proposed and the TAP development is also discussed. In addition, the TAS quality has influence on the rescheduling in real-life operations [5]. Therefore, the bottlenecks in a TAS should be detected and eliminated to improve the robustness of a TAS.

The approach to detecting bottlenecks in a TAS is proposed in the paper. A TAS is modeled by timed colored Petri nets and delay propagation under disturbances is collected, and then bottlenecks can be identified using departure delays. Finally, cases study based on a real station layout in China mainland railways and a thirty minutes timetable show that the approach enables to identify the bottlenecks in a TAS. The results would provide supports to TAS evaluation and robust TAP.

2. Track allocation schemes

Conflict free inbound & outbound routes to trains are formulated in a TAS. The set of trains is denoted by T , and the set of inbound and outbound routes are R^I and R^O respectively. The arrival and departure time for the train $t \in T$ is denoted by a_t and d_t respectively, and the inbound and outbound routes for the train t are i_t and o_t , and the job for the train t can be described by the five-tuple $\langle t, a_t, d_t, i_t, o_t \rangle$. So, the TAS including all the trains can be modeled by the following formula.

$$TAS = \bigcup_{t \in T} \{ \langle t, a_t, d_t, i_t, o_t \rangle \} \quad (1)$$

Although a feasible TAS ensures that there are no conflicts between any two jobs, the impact of a single job upon the TAS is neglected. In order to analyze the jobs in a TAS, its dynamic behaviors should be modeled first.

2.1. Concurrence and resources conflicts

The job $\langle t, a_t, d_t, i_t, o_t \rangle$ is composed of arrival and departure events j_t^I and j_t^O for the train t , and the set of arrival and departure events of all the trains is denoted by J^I and J^O respectively. A route is built up by linked track circuit sections (TC), and a job is composed of successive activities if occupancy to a TC for a train is considered an activity and the TCs in a route are released one by one. The set of TCs is G and $G(r)$ is the queue of the TCs in the route r with $r \in R = R^I \cup R^O$ and $G(r) \subset G$. The train for the job j is denoted by $t_j \in T$ for $\forall j \in J = J^I \cup J^O$ and $r_j \in R$ is the route for the job j , and $a(j, g)$ is the activity for the job j at the TC $g \in G(r_j)$ and it can be formulated by the five-tuple.

$$a(j, g) = \langle t_j, r_j, g, g_{t_j, r_j}^-, g_{t_j, r_j}^+ \rangle \quad (2)$$

The start and end times for the activity $a(j, g)$ are g_{t_j, r_j}^- and g_{t_j, r_j}^+ respectively, and it means that the TC g for the activity j is occupied in the time window $[g_{t_j, r_j}^-, g_{t_j, r_j}^+]$. The time windows for the jobs in a TAS can be obtained using the simulator in the literature [6].

Some activities are parallel if their time windows overlap. Fig.1 shows a simple station layout and there are six tracks, eight switches and ten TC. Supposed that there is a TAS including three jobs $\{j_{t_1}^I, j_{t_2}^O, j_{t_3}^O\}$ and their routs are $i_{t_1} = \langle g_1, g_2, g_6 \rangle$, $o_{t_2} = \langle g_2, g_1, g_3 \rangle$ and $o_{t_3} = \langle g_4, g_3 \rangle$. The activities for the jobs are listed in the Fig.2 with rectangles which represent the track section and its time window in the vertical and horizontal directions respectively. there are ten activities in the Fig.2 and the ones $a(j_{t_1}^I, g_1)$ and $a(j_{t_3}^O, g_3)$ are parallel.

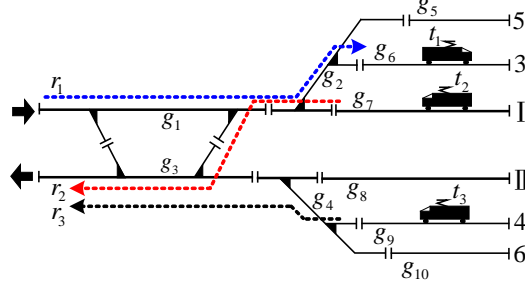


Fig.1 A simple station layout and three trains

There is potential conflict between two jobs if they would visit the same TC. In other words, the last one would be delayed if disturbance to the former one occurs, and the situation is referred to as resource conflicts.

2.2. Buffer times

There should be a certain interval between the time windows if there are resource conflicts for two jobs, and the time interval is buffer time. It expresses that the delay would propagate to the last job if the disturbance to the former one is larger than the buffer time between the two jobs. Two conditions shown in the formula (3)-(5) should be fulfilled for the existence of buffer time: the first is resource conflicts and the other expresses that the two jobs should be adjacent at a TC.

$$G(r_j) \cap G(r_k) \neq \emptyset, \forall k \in J, k \neq j \quad (3)$$

$$g_{t_k, r_k}^- > g_{t_j, r_j}^+, \forall k \in J, k \neq j, g \in G(r_j) \cap G(r_k) \quad (4)$$

$$g_{t_k, r_k}^+ < g_{t_m, r_m}^-, \forall m \in J, m \neq k \quad (5)$$

Take the jobs j and k for example: the resource conflicts is shown in the formula (3), and the formula (4) and (5) reflect that the activity $a(j, g)$ is followed by $a(k, g)$. Thus, the buffer time between jobs j and k at the TC g is $g_{t_k, r_k}^- - g_{t_j, r_j}^+$. Therefore, the buffer time between j and k is the minimum of those at all the TC and it is denoted by $b_{j,k}$:

$$b_{j,k} = \min \{ g_{t_k, r_k}^- - g_{t_j, r_j}^+ \}, \forall g \in G(r_j) \cap G(r_k) \quad (6)$$

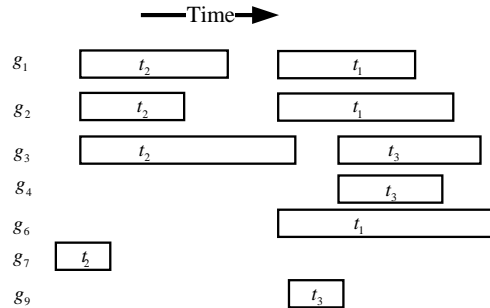


Fig.2 Time windows for the activities of three trains

In Fig. 2, the set of conflict resources for the jobs $j_{t_1}^I$ and $j_{t_2}^O$ is $\{g_1, g_2\}$ and the buffer time between them is located in the two activities at g_1 , and the buffer time for jobs $j_{t_2}^O$ and $j_{t_3}^O$

is at g_3 .

2.3. Bottlenecks detection

Delay propagation will occur if the disturbance is larger than the minimum buffer time in a TAS. When a job is disturbed, the delay propagation area is however determined not only by the disturbance and buffer times but by its neighbors. So, the bottlenecks are the jobs which caused the most serious delay propagation when each job is similarly disturbed. When a job j suffers the disturbance d , the train t is delayed for the time $l_t(j, d)$ and the sum of delays in a TAS is denoted by $Delays(j, d) = \sum_{t \in T} l_t(j, d)$. So, the job j satisfying formula (7) is a bottleneck.

$$j : \max_{j \in J} Delays(j, d) \quad (7)$$

It means that the delay caused by the disturbance (j, d) is more serious than that by (j', d) with $\forall j' \in J \wedge j' \neq j$. Thus, Perturbation analysis is a means of bottlenecks detection and it is necessary to establish the dynamic model of a TAS.

3. Timed Colored Petri nets

Colored Petri Nets (CPN) is particularly suitable for modeling concurrent and asynchronous behaviors and it has found successful applications on scheduling problem modeling^[7]. For TAS modeling, spatial and temporal conflicts and buffer times exist and hence the timed CPN is employed. The construction and validation of timed CPN model are now well supported by commercially available CPN software tools^[7]. The CPN symbols adopted below follows the syntax of CPN ML language^[7].

3.1. Introduction

The Petri nets can be modeled by the following formula.

$$CPN = (\Sigma, P, T, C, I, O, M_0) \quad (8)$$

The term $\Sigma = \{P, Q\}$ denotes the timed color set and P represents a train and Q a TC. The sets of places and transitions are represented by P and T respectively. The mapping $C : P \rightarrow \Sigma$ means the color of a place, and $I : (P \times T) \rightarrow \mathbb{N}$ is the input arc of the transition T and $O : (P \times T) \rightarrow \mathbb{N}$ is the output arc. $M_0 : P \rightarrow \mathbb{N}$ is the initial status of the model.

3.2. Concurrence and conflict modeling

With the basic CPN model shown in Fig.3, a train and a free TC are denoted by the timed Color Sets P and Q respectively. There are four places and two transitions in the model to represent the process of a train entering, staying and leaving a track section. The input places of the transition ‘Occupy’, denote a train $t1$ and a free track section $g1$; the output arc refers to the events that $t1$ occupies $g1$, and the value ‘90’ in the arc expression ‘ $t1@+90$ ’ represents the time-span (in seconds) of the event of occupation. The place ‘ $g1$ busy’ shows the state of the track section being occupied by the train $t1$. The transition ‘Release’ denotes train departure from the track section which becomes ‘free’ afterwards and thus ready to be occupied by the subsequent train.

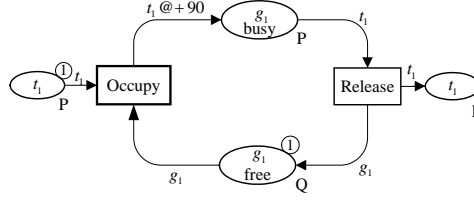


Fig.3 A colored Petri nets model for an activity

The resource conflict between jobs $j_{t_1}^I$ and $j_{t_2}^O$ is illustrated in Fig. 4 and the occupation times of the events are neglected for simplicity. The events “Occupy” and “Leave” represent the occurrence of the activities $a(j_{t_1}^I, g_1)$ and $a(j_{t_2}^O, g_1)$ respectively, and the place “g1 free” is the input conditions of the two activities. However, the TC g_1 only can serve one activity simultaneously and this reflects the resource conflict between the two jobs.

From the transition ‘Release g1’, the next event to be triggered (either to place ‘g2 busy’ or ‘g3 busy’) depends on the ‘input’ to the transition. To ensure an unambiguous model, a variable x is introduced in the arc inscription from ‘g1 busy’ to ‘Release g1’. The Color Set of x is P as defined in Fig.3, representing a train at the place ‘g1 busy’. In addition, the out-going arcs from ‘Release g1’ are defined according to the value of x . The transition ‘Release g2’ is specified similarly.

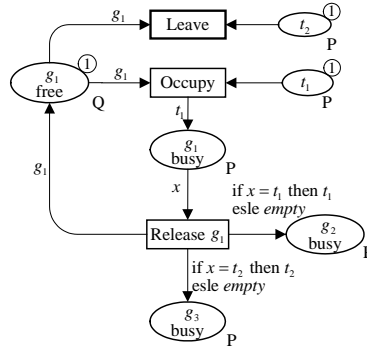


Fig.4 Two activities at the same track section

Station layout and service timetable impose spatial and temporal constraints on the TAP. A timed CPN model is proposed to encapsulate both constraints in a systematic depiction while the buffer time is also embedded implicitly. This model enables to collect the delay information at each place upon service disturbances. Furthermore, for a TAS with large-scale station layout and more trains, the hierarchical modeling CPN ML language can be used to establish large-scale Petri nets.

4. Cases study

The station layout linking six directions from A to F is shown in Fig. 5 and Table 1 lists a TAS with eight trains for the duration of approximately thirty minutes. The arrival and departure routes are represented by the symbols ① ... ⑥ shown in Fig.5. The path ① means the arrival route of the trains from the direction E and the departure route to E along the main line is neglected to make the graph easy to be observed.

The time windows of the activities in the input TAS can be obtained using the simulator in the literature [6] and then the buffer times for each job are achieved. Only the buffer times lower than 180 seconds are shown in Table 2 as the input service disturbances are less than 3 minutes. For example, the first row in Table 2 means that the buffer time between ‘train 1’ and ‘train 2’ departures is 40 seconds. The minimum buffer time is 21 seconds and it is between the ‘train 5’ and ‘train 6’ arrivals. The following two ones are 30 and 31 seconds which stay between ‘train 4’ and ‘train 5’ arrivals and between ‘train 7’ and ‘train 8’ departures respectively.

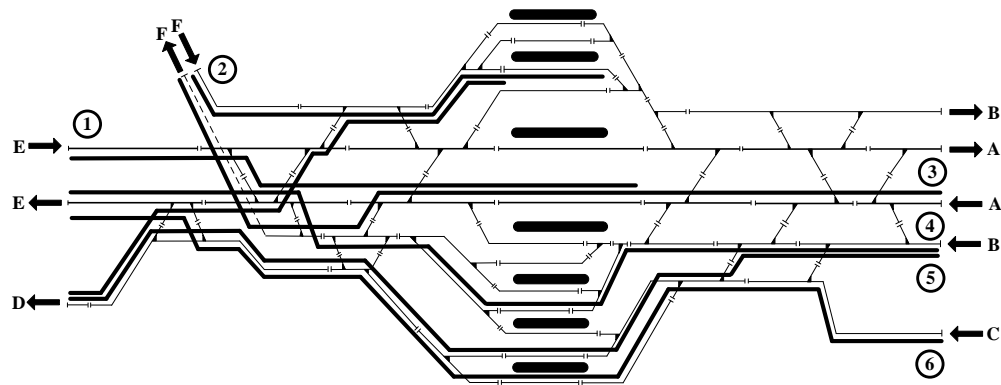


Fig.5 The station layout and paths for trains

Table 3 shows the sum of departure delays of all the trains when a train is disturbed. For example, when ‘train 1’ suffers a service disturbance of 60 seconds, the total departure delay of the eight trains is 80 seconds, and when ‘train 1’ suffers 80 seconds, the total departure delay is 120 seconds. It is assumed that there is no buffer time for trains’ operations dwelling on platform tracks and an arrival delay will result in the same departure delay, so only the departure delays are collected in the Petri nets model of the TAS.

Table1 A track allocation scheme with eight trains

Train ID	Arrival Time	Departure Time	Inbound Direction	Outbound Direction	Paths
1	8:03:00	8:05:00	A	F	③
2	8:06:20	8:08:20	B	E	④
3	8:09:00	8:14:50	B	D	⑤
4	8:14:20	8:15:20	E	E	①
5	8:15:06	8:18:16	F	D	②
6	8:17:00	8:20:40	C	E	⑥
7	8:19:35	8:23:15	B	E	④
8	8:23:08	8:26:55	E	E	①

Table2 Buffer times between jobs for eight trains

The Previous train	Jobs	The following train	Jobs	Buffer time (seconds)
1	Arrival	2	Departure	40
2	Arrival	3	Arrival	57
2	Departure	4	Arrival	44
3	Departure	5	Departure	100
4	Departure	5	Departure	30
5	Departure	8	Arrival	43
5	Departure	6	Departure	21
6	Departure	7	Departure	33
7	Departure	8	Departure	31
8	Arrival	7	Departure	42

From Table 3, the most serious departure delays under service disturbances are caused by the ‘train 5’ and the following two are ‘train 4’ and ‘train 2’, and the sequence is recorded by $\langle 5, 4, 2 \rangle$. From Table 2, the least three buffer times stay between trains 5, 4, 7 and their following trains, and the sequence is $\langle 5, 4, 7 \rangle$ here. The two sequences are not exactly the same and it means that the caused delays are not completely identical to the order of buffer times. The argument is proved again with the continuous increase in disturbances: when the service disturbances are 150 and 180 seconds, the sequence of trains which caused the largest three delays is $\langle 2, 4, 5 \rangle$ and it is completely different to the sequence $\langle 5, 4, 7 \rangle$ which indicates the least three buffer times.

The sum of each row in Table 3 denotes the total departure delays when service disturbances are separately imposed on each train, and the largest two delays are 2067 and 2063 seconds. Since the difference between the two values is not various obvious, it can be concluded that there are two bottlenecks in the TAS and they are ‘train 2’ and ‘train 4’.

At the same time, the delay propagation is affected not only by the buffer times but by the sequence of trains’ jobs, and there is more propagation space for the disturbance imposed on the earlier jobs. However, the approach can still be adopted to perform perturbation analysis of a TAS and the order of trains’ jobs can be ignored for the evaluation of a group of TAS with the same station layout and the timetables.

Tables 3 Departure delays of the trains under disturbances

Train ID	Disturbances (seconds)					
	60	80	100	120	150	180
1	80	120	176	242	377	562
2	79	145	230	333	535	745
3	60	80	100	140	236	382
4	99	166	262	362	512	662
5	122	202	282	362	482	602
6	87	143	203	263	353	443
7	89	129	169	209	269	329
8	78	118	158	198	258	318

Although the largest total delay is caused by the disturbances imposed on the ‘train 2’, it should be emphasized that it is not only due to routes allocation to ‘train 2’ but the neighbors of

‘train 2’. From Table 2, the job of ‘train 2’ arrival is followed by ‘train 3’ and ‘train 4’ arrivals, and the corresponding buffer times are 57 and 44 seconds respectively. It means that the four jobs should be considered together when the local adjustment of the TAS is needed.

5. Results and Discussion

Buffer time is rarely addressed in the previous studies to formulate the TAP and the robustness of TAS is ignored. Given a TAS, bottlenecks detection and local improvement can support to tackle the robustness. A timed colored Petri nets model of TAS is proposed to perform perturbation analysis under service disturbances. Cases study based on a station layout linking six directions and a timetable for the duration of about 30 minutes shows that 602 seconds departure delay is caused when a disturbance of 180 seconds is imposed on the job where the minimum buffer time (21 seconds) is located, and the largest delay is 745 seconds under the same disturbance and the corresponding buffer time is 57 seconds. There are two bottlenecks in the TAS by comparing the departure delays caused by each job under service disturbances.

The results suggest two meanings: bottlenecks detection can provide supports to TAS evaluation and robustness optimization by reallocating the buffer times. In addition, the bottlenecks can be provided to support the integrated methodology for robust railway timetabling.

References:

- [1] P.J. Zwaneveld, L.G. Kroon, et al. Routing Trains through a Railway Station based on a Node Packing Model [J]. *European Journal of Operational Research*, 2001, 128(1): 14-33.
- [2] SHI Feng, Xie Chu-nong, YU Gui-fang. Optimization Methods for Arranging Train Routes in Station Bottleneck [J]. *Journal of the China Railway Society*, 2004, 26(4): 5-9.
- [3] S. Lan, C. Barnhart, J.P. Clarke. Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions [J]. *Transportation Science*, 2006, 40(1): 15–28.
- [4] MAO Bao-hua, WANG Bao-shan, XU Bin, et al. On Integrated Train Operational Scheme Drawing-up Platform of China [J]. *Journal of Transportation Systems Engineering and Information Technology*, 2009, 9 (2): 27-37.
- [5] D. Andrea, P. Dario, et al. Assessment of Flexible Timetables in Real-time Traffic Management of a Railway Bottleneck [J]. *Transportation Research Part C*, 2008, 16 (2): 232–245.
- [6] MAO Bao-hua, HE Tian-jian, et al. A General-purposed Simulation System on Train Movement [J]. *Journal of the China Railway Society*, 2000, 22(1): 1-6.
- [7] K. Jensen, L. Kristensen, L. Wells. Colored Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems [J]. *International Journal on Software Tools for Technology Transfer*, 2007, 9(3): 213-254.